

## Desain dan Simulasi Sistem Kendali PID Pada AGV(Automated Guided Vehicle) Pengikut Garis

Moch. Iskandar Riansyah<sup>1</sup>, Isa Hafidz<sup>2</sup>, Dimas Adiputra<sup>3</sup>, Lora Khaula Amifia<sup>4</sup>, Anifatul Faricha<sup>5</sup>

<sup>1,2,3,4,5</sup>Teknik Elektro, Fakultas Teknik Elektro, Institut Teknologi Telkom Surabaya

Email: <sup>1</sup>iskandar@ittelkom-sby.ac.id

**Abstract.** Effectiveness and efficiency in the field of Logistics is very important. One solution to improve this is to automate the logistics process with robotics technology called AGV (Automated Guide Vehicle). In this study AGV will be developed which is driven by an algorithm so that it can be followed by a white path when it is discussed from one place to another in the logistics process. The access system used is the PID control (Proportional, Integral, Derivative), with hardware support that uses the RGB-D camera as an AGV navigation tool to support and access the track. PID Control System to keep AGV on the track created. Testing is done by comparing the values of parameters both P, PD, and PID in order to get the parameters that produce the best response so that AGV is able to move through the white line trajectory. AGV is designed and simplified in an open source software known as the Gazebo simulator and ROS (Robot Operating System). The navigation process uses a line detection algorithm by making filtering and thresholding to determine the center point when drawing a white line. Given the center point, AGV moves to follow the path pattern by comparing it to a predetermined reference point

**Keywords:** PID Control, AGV (Automated Guided Vehicle), RGB-D, Gazebo Simulator, ROS(Robot Operating System)

**Abstrak.** Efektivitas dan efisiensi di bidang Logistik menjadi hal yang sangat penting. Salah satu solusi yang dilakukan untuk meningkatkan hal tersebut adalah bagaimana mengotomasi proses logistik dengan teknologi robotika yang disebut dengan AGV (Automated Guide Vehicle). Pada penelitian ini akan dikembangkan AGV yang dikendalikan dengan suatu algoritma agar dapat mengikuti lintasan berupa garis berwarna putih pada saat bertugas memindahkan material dari satu tempat ke tempat lain di dalam proses logistik. Sistem kendali yang digunakan adalah kendali PID (Proportional, Integral, Derivative), dengan dukungan perangkat keras berupa kamera RGB-D sebagai alat navigasi AGV untuk mendeteksi dan mengikuti lintasan. Sistem Kendali PID berfungsi untuk menjaga agar AGV tetap berada pada lintasan yang dibuat. Pengujian dilakukan dengan membandingkan nilai-nilai parameter baik kendali P, PD, dan PID agar mendapatkan parameter yang menghasilkan respon paling baik sehingga AGV mampu bergerak mengikuti lintasan garis berwarna putih. AGV telah dirancang dan disimulasikan dalam sebuah perangkat lunak open source yang dikenal sebagai simulator Gazebo dan ROS (Robot Operating System). Proses navigasi menggunakan algoritma deteksi garis dengan tahapan melakukan filtering dan thresholding untuk menentukan titik pusat moment gambar lintasan garis putih. Dengan adanya titik pusat tersebut, AGV bergerak mengikuti pola lintasan dengan membandingkannya terhadap titik referensi yang telah ditentukan

**Kata Kunci:** Kendali PID, AGV (Automated Guided Vehicle), RGB-D, Simulator Gazebo, ROS(Robot Operating System)

### 1. Pendahuluan

Dalam beberapa dekade ini, perkembangan teknologi robotika begitu pesat. Teknologi robotika telah banyak dipergunakan dalam berbagai bidang aplikasi untuk membantu aktivitas manusia menjadi lebih cepat, tepat, mudah, dan aman. Terutama bidang yang saat ini menjadi sangat trend yaitu logistik, dimana perpindahan barang dari satu tempat ke tempat lain begitu cepat dan menjadi kebutuhan setiap orang lintas wilayah. Di bidang logistik, efisiensi dan efektifitas menjadi suatu hal penting dan mendasar yang harus dikelola dengan baik. Efisiensi ini berkaitan dengan meningkatkannya produktivitas dan turunnya biaya yang dikeluarkan untuk distribusi barang di dalam logistik, sebagai contoh adalah biaya pekerja yang cukup besar untuk mendistribusikan barang dari

satu tempat ke tempat lainnya. Sedangkan efektivitas, berkaitan dengan ketepatan dan kecepatan agar dalam hal mendistribusikan barang terjadi adanya pengurangan kesalahan yang dapat menghambat produktivitas. Apabila kedua hal tersebut tidak dikelola dengan baik, maka akan menimbulkan biaya yang dikeluarkan lebih besar dan waktu proses dalam logistik juga berjalan lebih lama sehingga membuat kualitas pelayanan distribusi terhadap pelanggan atau pengguna akan semakin menurun. Logistik merupakan salah satu hal yang tidak dapat terlepas dari dunia industri dan perdagangan. Hingga saat ini, industri dan gudang telah menggunakan suatu teknologi robotika yang disebut dengan AGV (Automated Guided Vehicle) untuk mengoptimalkan sistem penanganan material dan juga sebagai alternatif untuk menyelesaikan masalah yang disebutkan berhubungan dengan masalah sistem pendistribusian. Keuntungan utama dari AGV adalah adanya pengurangan tenaga kerja sehingga biaya yang dikeluarkan dapat diturunkan dan adanya peningkatan fleksibilitas pada perencanaan pendistribusian barang (Qi, Zhou and Yang, 2016). Pada suatu penelitian yang telah dilakukan (Li, Yan and Li, 2018), menyatakan bahwa AGV sering kali digunakan pada kegiatan manufaktur dan melakukan tugas transportasi pada (internal dan eksternal) area transportasi, transshipment dan distribusi. ITS (Sistem Transportasi Cerdas) memiliki potensi meningkatkan keselamatan dan produktivitas pada operasi pelabuhan, meningkatkan nilai ekonomis pengiriman antar moda operasi, di mana AGV memiliki peran penting. AGV merupakan kendaraan yang dilengkapi dengan perangkat keras, perangkat lunak, dan sensor serta aktuator yang bertanggung jawab mengarahkan pergerakan sesuai dengan rute yang telah direncanakan. Perbedaan yang mendasar adalah bahwa AGV dipandu dan digerakkan secara otomatis tanpa menggunakan seorang pengemudi. Orientasi pergerakan AGV biasanya dipandu menggunakan sistem optik baik laser dan ataupun kamera, sistem magnetik atau radio, dan juga menggunakan penggabungan beberapa sistem yang menerapkan sistem fusi data sensor. Di lingkungan industri, jenis AGV yang paling banyak digunakan adalah sistem optik atau sistem magnetik. Seperti pada penelitian (Tran et al., 2018) yang mengembangkan suatu AGV yang digunakan untuk mengambil dan menyimpan barang di gudang dengan lintasan berupa garis hitam. AGV ini dilengkapi dengan suatu sensor optik berupa infrared sebagai sensor deteksi garis. Sensor bekerja dengan menangkap data pantulan cahaya infrared yang dipancarkan ke garis berwarna hitam yang kemudian di olah dengan algoritma tertentu agar robot dapat menuju lokasi pengambilan atau pun penyimpanan barang di gudang yang diinginkan oleh pengguna. Beberapa penelitian juga telah menggunakan sebuah kamera sebagai pengganti sensor optik infrared karena kemudahan pemasangan dan integrasi dengan perangkat lain (Gomes et al., 2016). Untuk model yang cukup lama digunakan adalah model magnetik dengan keunggulan tidak dipengaruhi oleh cahaya namun dari memiliki kesulitan dari segi pemasangan infrastruktur pendukungnya yaitu lintasan magnetik yang dibangun (Qi, Zhou and Yang, 2016). Namun, tren lain yang saat ini telah dikembangkan adalah AGV dengan navigasi menggunakan QR Code dimana robot akan mengikuti pola lintasan QR Code yang dipasang. Penggunaan QR Code saat diuntungkan dari segi kemudahan pemasangan infrastruktur lintasan yang akan dilalui (Juntao et al., 2015). Salah satu tujuan dari penelitian ini adalah merancang dan mendesain AGV menggunakan kamera RGB-D sebagai sensor yang mengintegrasikannya ke metode kendali, seperti PID (Proportional-Integral-Derivative) dengan simulator Gazebo dan ROS. Penggunaan simulasi ini dimaksudkan agar penerapan ke dalam sistem nyata akan lebih mudah, karena platform ROS pada simulasi memiliki pendekatan implementasi yang tidak jauh dari kenyataannya.

## **2. Tinjauan Pustaka**

### **2.1. Perangkat Lunak Simulasi**

ROS pada simulasi memiliki pendekatan implementasi yang tidak jauh dari kenyataannya. GAZEBO Simulator adalah sebuah simulator yang digunakan untuk simulasi berbagai platform dan aplikasi robot (lingkungan, jenis robot, sensor dan aktuator) yang semuanya dirancang dengan sifat fisik dan struktural (Takaya et al., 2016). Sketchup adalah software yang digunakan untuk rendering dan pemodelan 3D. Suatu perangkat lunak open source untuk mengontrol robot di lingkungan simulasi disebut dengan ROS (Robot Sistem Operasi) diterapkan dengan menerima data dari kamera dan mengirim ke algoritma kendali. ROS telah memiliki package untuk mobile robot jenis Ackermann seperti AGV yang bisa dikembangkan untuk simulasi. Selain itu kompatibilitas sistem ROS yang

dikembangkan di dalam simulasi dapat diimplementasikan langsung pada kendaraan AGV sebenarnya. Algoritma kendali dibuat dalam bahasa pemrograman python dan OpenCV.

## 2.2. Sistem Kendali PID

ROS pada simulasi memiliki pendekatan implementasi yang tidak jauh dari Sistem Kendali PID pada dasarnya merupakan sistem kendali yang memiliki 3 komponen yang menyusunnya, yaitu Proporsional, Integral, dan Derivative(Sampath, Perera and Dassanayake, 2014). Persamaan nilai keluaran dari kendali PID, dirumuskan sebagai:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

Persamaan (1) menjelaskan bahwa nilai keluaran  $u(t)$ , merupakan jumlah dari *gain proportional* ( $K_p$ ), *gain integral* ( $K_i$ ), dan *gain derivative* ( $K_d$ ) yang masing-masing dipengaruhi oleh *error* ( $e$ ) dalam selang waktu ( $t$ ) tertentu.

### 1) Proportional Control

Kendali proporsional memiliki output yang sebanding atau proporsional dengan besar sinyal kesalahan (selisih antara nilai yang diinginkan dengan nilai aktualnya, *error*). Setiap perubahan pada sinyal input menyebabkan sistem langsung mengeluarkan sinyal keluar sebesar konstanta pengalinya. Persamaan kendali proporsional dirumuskan sebagai :

$$u(t) = K_p e(t) \quad (2)$$

$K_p$  merupakan gain proportional,  $e$  adalah error, dan  $u$  merupakan nilai keluaran relatif terhadap waktu ( $t$ )

### 2) Integral Control

Pengontrol integral berfungsi untuk menghilangkan *steady-state error* menjadi nol. Jika sebuah plant tidak mempunyai unsur integrator ( $1/s$ ), pengontrol proporsional tidak mampu menjamin output sistem akan tepat sesuai respon yang diinginkan, sehingga dibutuhkan pengontrol integral. Persamaan kendali integral dirumuskan sebagai:

$$u(t) = K_i \int_0^t e(t) dt \quad (3)$$

$K_i$  merupakan gain integral,  $e$  adalah *error*, dan  $u$  merupakan nilai keluaran relatif terhadap waktu ( $t$ )

### 3) Derivative Control

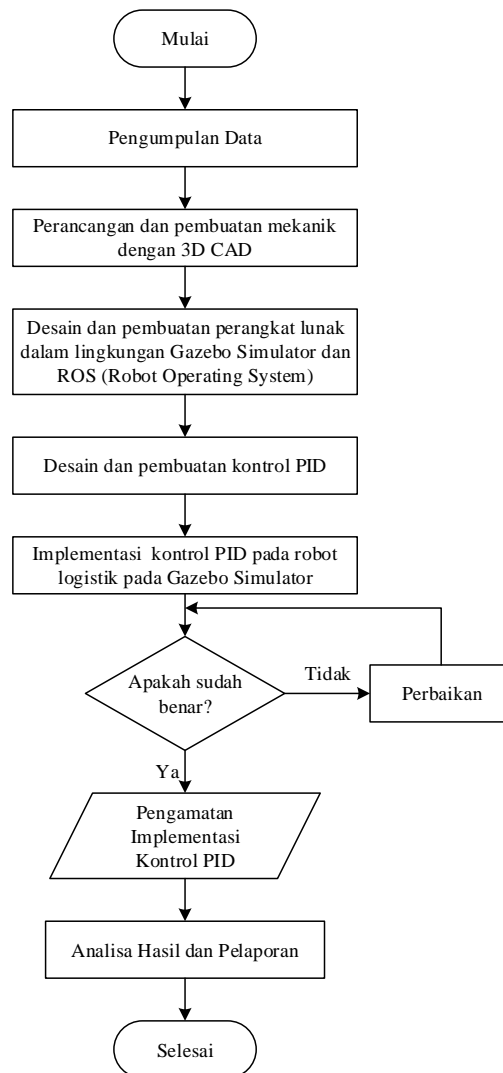
Besar output dari pengontrol derivatif memiliki sifat seperti operasi diferensial pada umumnya. Pengontrol derivatif menggunakan kecepatan perubahan sinyal kesalahan sebagai parameter pengontrol. Apabila tidak ada perubahan sinyal error, maka output dari kendali derivative tidak akan berubah. Persamaan kendali derivatif dirumuskan sebagai :

$$u(t) = K_d \frac{de(t)}{dt} \quad (4)$$

$K_d$  merupakan gain derivatif,  $e$  adalah error, dan  $u$  merupakan nilai keluaran relatif terhadap waktu ( $t$ ).

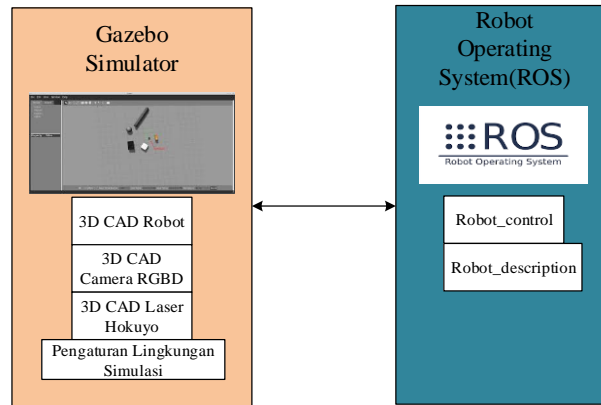
### 3. Metode Penelitian

Pada bagian ini, akan menjelaskan mengenai perancangan dan pembuatan simulasi AGV dengan kendali PID pada lingkungan simulator Gazebo yang diintegrasikan dengan ROS (Robot Operating System). Pada tahap awal, dilakukan pengumpulan informasi terkait dengan perkembangan AGV. Selanjutnya, dilakukan proses desain AGV yang akan diimplementasikan pada Simulasi dengan platform Gazebo dan ROS. Setelah proses ini selesai, maka dilakukan pengujian secara langsung pada simulasi dengan menerapkan algoritma kendali PID. Kemudian hasil proses pengujian pada simulasi akan dianalisa dan dilaporkan dalam bentuk tulisan . Berikut adalah tahapan yang dilakukan seperti pada gambar 1



Gambar 1. Diagram Alir Penelitian

Dari gambar 2 blok diagram diatas bahwa sistem dibangun dengan mengintegrasikan dua perangkat lunak yaitu Gazebo simulator dan ROS (Robot Operating System). Gazebo simulator akan menangani proses simulasi AGV yang memerlukan gambar 3D benda yang akan disimulasikan pada lingkungannya. Sedangkan untuk blok diagram sistem secara keseluruhan adalah seperti pada Gambar 2. berikut ini.

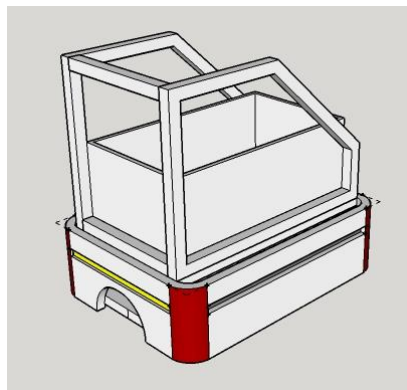


Gambar 2. Blok Diagram Sistem Simulasi

Pada AGV dilakukan penggabungan beberapa 3D CAD yaitu mekanik robot logistik, kamera RGB-D dan Laser Hokuyo sebagai navigasi, dan Lingkungan yang diinginkan berupa lintasan garis berwarna putih. Agar 3D CAD tersebut dapat digerakkan dan dikendali di lingkungan simulator Gazebo, maka diperlukan sebuah sistem operasi opensource yaitu, ROS (Robot Operating System). Di dalam ROS inilah, semua algoritma yang didesain dan dibuat ditanamkan, yang kemudian akan saling berkomunikasi dengan lingkungan simulasi.

### 3.1. Perancangan Mekanik AGV

Desain konseptual dari AGV telah dibuat seperti yang terlihat pada Gambar 3. Desain dibuat menggunakan software Sketch Up yang kemudian hasilnya dikonversi dalam bentuk .dae karena itu merupakan spesifikasi yang diinginkan oleh simulator Gazebo.



Gambar 3. Desain Mekanik 3 Dimensi Robot Logistik

Desain dibuat sedemikian rupa memiliki tempat untuk mengangkat barang karena robot ini akan difungsikan sebagai robot untuk mendistribusikan barang dari satu tempat proses ke tempat proses lainnya di dalam industri manufaktur.

### 3.2. Komponen Pendukung

#### 1) Kamera RGB-D

Sebuah kamera simulasi terpasang dibagian tengah badan robot, yang menghadap ke bawah berfokus pada lintasan yang akan dilalui. Kemudian gambar yang diambil akan dianalisis untuk memperkirakan pergerakan yang akan dilakukan robot mengikuti lintasan garis berwarna putih yang dibuat.

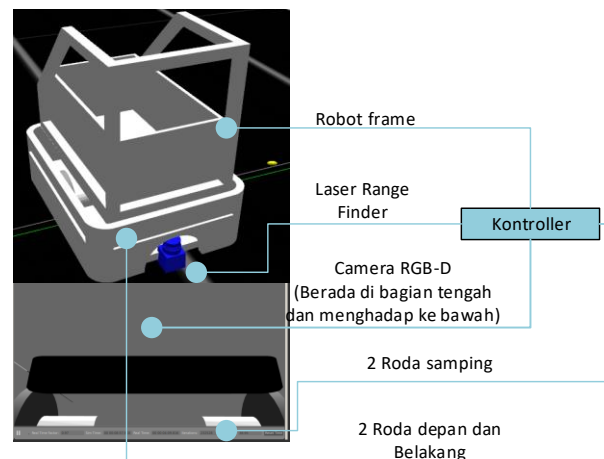
#### 2) Laser Range Finder

Sebuah Laser Range Finder Hokuyo simulasi terpasang di depan mekanik AGV yang dimaksudkan sebagai sensor yang mendeteksi adanya halangan atau rintangan. Dengan algoritma

tertentu yang mengolah data dari sensor tersebut, robot akan dapat menghindari dari halangan atau rintangan yang ditemui. Namun, pada simulasi ini sensor laser range finder belum difungsikan karena diperuntukkan untuk eksperimen selanjutnya dan tujuan yang mendasar adalah masih pada penggunaan kamera untuk navigasi sehingga lingkungan simulasi dibuat tanpa adanya halangan

### 3.3. Desain Sistem Robot Logistik

Desain sistem seperti yang telah ditunjukkan pada Gambar. 3 terdiri dari kamera RGB-D, Laser Range Finder Hokuyo, Badan Robot Logistik, dua roda linear dan dua roda bebas.



Gambar 4. Desain Sistem Robot Logistik

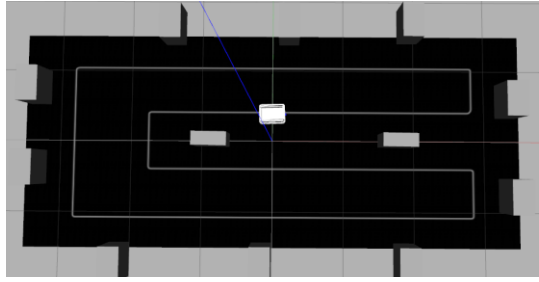
Berdasarkan pengolahan data dari kamera RGBD kemudian perintah ke motor kiri dan kanan dikirim untuk menavigasi robot di sepanjang garis putih.

### 3.4. Lingkungan Pengujian Simulasi

Adapun lingkungan pengujian simulasi dibuat menggunakan software Sketchup dengan memasukkan file .dae nya ke dalam lingkungan simulator gazebo berbentuk world yang kemudian akan dijalankan dengan menggunakan ROS (Robot Operating System). Lingkungan simulasi ini, adalah berupa suatu lintasan garis berwarna putih dengan ukuran lebar 2 cm berbentuk huruf U tanpa adanya halangan atau rintangan. Sebelumnya, telah dibuat bentuk image dari lintasan seperti pada Gambar 5. Sedangkan Gambar 6 menunjukkan lingkungan untuk pengujian simulasi setelah image dimasukkan pada simulator Gazebo.



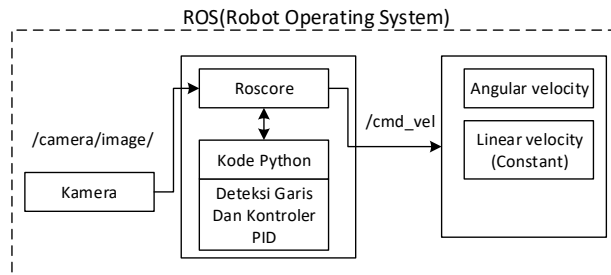
Gambar 5. Desain Gambar Lintasan Garis Robot Logistik



Gambar 6. Lintasan Garis AGVPada Simulator Gazebo

### 3.5. Perancangan Perangkat Lunak

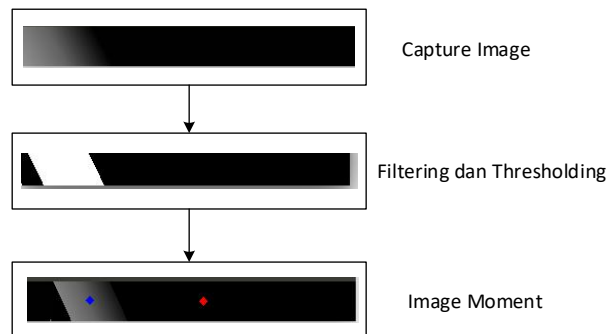
Konsep dari perancangan perangkat lunak secara garis besar dibuat dengan menggunakan platform Robot Operating System (ROS). ROS menawarkan suatu kemudahan dalam mengintegrasikan suatu sistem sehingga komunikasi data antar satu perangkat ke perangkat lain telah dapat ditangani oleh sebuah perintah yang disebut dengan roscore sebagai pengendali dan pemroses utama yang ada di dalam ROS. Sistem yang dirancang, adalah sistem AGV yang mampu mengikuti suatu garis berwarna putih menggunakan sebuah kamera RGB-D. Kamera ini akan menangkap setiap gambar selama bergerak secara realtime setiap 30 fps. Selanjutnya, akan diproses menggunakan algoritma deteksi lintasan untuk dapat mengikuti pola lintasan yang dilalui dengan mengendalikan kecepatan linier dan kecepatan sudut robot. Agar dapat dikendalikan secara *smooth* mengikuti lintasan garis maka diterapkanlah suatu kendali PID pada robot. Berikut adalah blok diagram sistem perangkat lunak ROS (Robot Operating System) yang ditunjukkan pada Gambar 7.



Gambar 7. Blok Diagram Sistem Perangkat Lunak

#### 1) Sistem Deteksi Lintasan

Sistem deteksi lintasan dibuat untuk membantu robot melakukan pergerakan mengikuti lintasan berupa garis berwarna putih pada simulator Gazebo. Terdapat beberapa tahapan yang dilakukan agar robot mampu mengidentifikasi garis warna putih yaitu proses menangkap gambar (Capture image), proses pemfilteran dan threshold, serta menentukan nilai moment dari gambar hasil filter dan threshold.

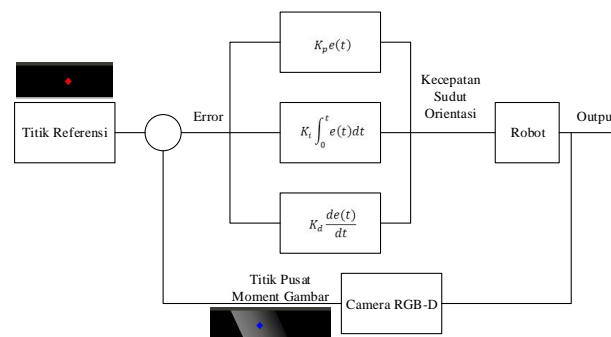


Gambar 8. Blok Diagram Proses Deteksi Garis

Gambar yang ditangkap kamera RGB-D akan dikonversi menjadi grayscale image dan dikompresi ukurannya menjadi 200 x 450 pixel. Gambar grayscale tersebut kemudian akan diubah dari tipe RGB menjadi tipe HSV. Dengan mengetahui nilai HSV garis berwarna putih baik batas atas maupun batas bawahnya, maka kita dapat melakukan proses pemfilteran. Selanjutnya, gambar yang telah difilter akan dilakukan thresholding sehingga menghasilkan sebuah gambar biner. Gambar biner inilah yang kemudian akan dicari nilai momentnya sehingga bisa menghasilkan suatu titik pusat yang akan dibandingkan jaraknya dengan titik referensi. Sistem deteksi lintasan ini, dibuat dengan menggunakan Bahasa pemrograman python dan OpenCV dalam lingkungan Robot Operating System (ROS).

## 2) Sistem Kendali PID

Sistem kendali yang digunakan pada penelitian ini adalah sistem kendali PID (Proportional, Integral, dan Derivative). Dimana kendali PID ini digunakan untuk mengendalikan arah orientasi robot terhadap lintasan garis berwarna putih yang dilalui.

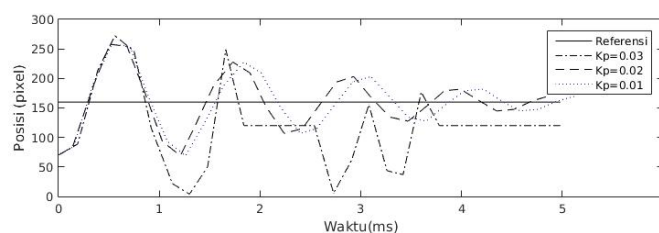


Gambar 9. Blok Diagram Sistem Kendali PID

Posisi titik pusat gambar atau disebut dengan moment image yang diperoleh dari hasil tangkapan sensor kamera akan dibandingkan dengan posisi titik referensi yang telah ditentukan pada sebuah layar buatan. Melalui hal tersebut, akan diperoleh nilai error antara kedua posisi titik yang kemudian akan dihitung dengan PID agar menghasilkan suatu sinyal kendali berupa kecepatan sudut orientasi. Kecepatan sudut orientasi ini yang akan menggerakkan robot sesuai dengan alur lintasan garis berwarna putih pada simulator Gazebo

## 4. Hasil dan Pembahasan

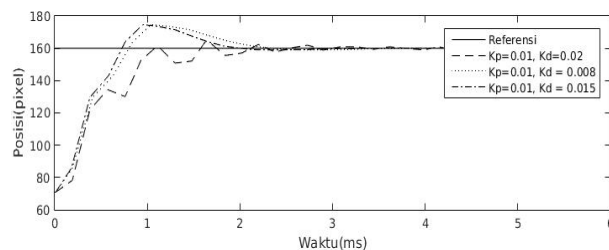
Pada bagian ini akan membahas mengenai proses implementasi dan pengujian sistem secara keseluruhan pada simulator Gazebo. Robot yang akan dijalankan pada simulasi, diatur memiliki kecepatan konstan  $v = 0.18$  m/s. Dengan kecepatan tersebut robot akan melintasi lintasan berbentuk U yang telah dirancang sebelumnya di dalam simulator. Untuk mengetahui kinerja sistem robot dalam mengikuti pola lintasan garis putih yang dilewati sehingga data yang diperoleh dapat dianalisa, maka proses pengujian dilakukan pada sistem kendali PID yang diterapkan. Tahap pengujian awal adalah dengan menerapkan kendali P pada sistem deteksi lintasan robot.



Gambar 10. Grafik Respon Kendali Proporsional

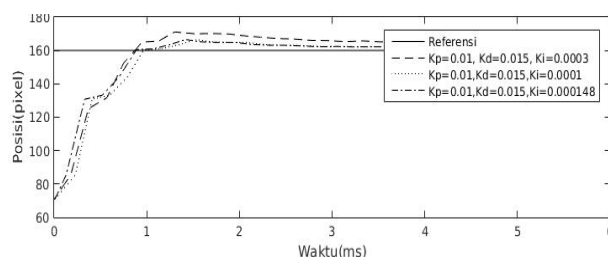


Pengujian ini dilakukan untuk mengetahui kecepatan respon kesesuaian posisi titik pusat moment gambar lintasan yang ditangkap oleh kamera terhadap titik referensi yang ditentukan. Dari Gambar 10, dapat dilihat bahwa nilai  $K_p = 0.03$ ,  $K_p=0.02$ , dan  $K_p=0.01$  memiliki kecepatan respon yang hampir sama. Namun, tingkat osilasi yang dihasilkan dengan nilai  $K_p=0.03$  cenderung besar sehingga dapat diamati bahwa sistem mengalami ketidakstabilan. Sedangkan untuk nilai  $K_p=0.02$  dan  $K_p=0.01$  tingkat osilasi mengecil dan sistem saat kisaran waktu 5 ms dapat mencapai referensi sehingga kesalahan terhadap referensi akan mengecil.



Gambar 11. Grafik Respon Kendali PD (Proporsional, Derivative)

Tahap pengujian yang kedua adalah dengan menambahkan kendali Derivative pada sistem. Hal ini dilakukan karena pada pengujian awal, penggunaan kendali proporsional belum dapat bekerja maksimal karena sistem masih belum stabil yang ditandai dengan adanya osilasi respon. Sehingga, untuk mengatasi osilasi akibat overshoot, maka pengujian selanjutnya dilakukan dengan menambah kendali derivative yang mampu mengendalikan dan menekan overshoot yang terjadi pada sistem deteksi lintasan robot. Pada Gambar 11, dapat dilihat bahwa dengan penambahan  $K_d$  sebesar 0.02, 0.008, dan 0.015 dapat mengurangi adanya overshoot dan nilai kesalahan terhadap referensi sehingga sistem dapat lebih cepat mencapai referensi. Namun, untuk respon sistem yang memiliki eror lebih kecil ditunjukkan dengan nilai  $K_d=0.02$  dan  $K_d=0.015$ .



Gambar 12. Grafik Respon Kendali PID (Proporsional, Integral, dan Derivative)

Tahap selanjutnya adalah dengan menambahkan kendali integral pada sistem sehingga dapat memperkecil lagi nilai error steady state dari respon sistem. Terlihat pada pengujian kendali PD overshoot dan kesalahan masih dimiliki oleh sistem dalam mendeteksi lintasan garis putih. Pada Gambar 11, dengan adanya penambahan kendali Integral dengan  $K_i$  sebesar 0.003, 0.001, dan 0.00148 membuat overshoot dan kesalahan yang terjadi semakin kecil sehingga menyebabkan sistem berjalan terkendali dengan baik. Dengan demikian, pergerakan robot mengikuti lintasan pada simulator gazebo akan berjalan dengan maksimal dan stabil karena kesalahan pada posisi moment gambar lintasan garis putih terhadap referensi sangat kecil.

## 5. Penutup

### Kesimpulan

Dari ketiga tahapan pengujian yang telah dilakukan, apabila ingin meningkatkan kecepatan respon sistem dari posisi titik pusat moment terhadap titik referensi maka dapat dilakukan dengan menambahkan komponen nilai  $K_p$ . Parameter  $K_d$  memiliki pengaruh yang besar untuk menekan adanya overshoot pada respon sistem karena proses kenaikan atau penurunan posisi titik mengalami

perubahan yang cepat. Dengan penambahan kendali Integral maka hasil pemilihan parameter konstanta PID yang tepat untuk sistem yaitu  $K_p=0.01$ ,  $K_i=0.015$  dan  $K_d=0.000148$  yang memiliki overshoot 25% dan sudah stabil terhitung dari setelah 2 ms memiliki steady state error sebesar 0,52%.

### Ucapan Terima Kasih

Peneliti mengucapkan terimakasih kepada Dekan Fakultas Teknik Elektro dan rekan-rekan di Institut Teknologi Telkom Surabaya yang sudah memberi masukan dan dukungan sehingga dapat menyelesaikan penelitian ini.

### Daftar Pustaka

- Gomes, M. V. *et al.* (2016) 'PID control applied on a line-follower AGV using a RGB camera', *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 194–198. doi: 10.1109/ITSC.2016.7795553.
- Juntao, L. *et al.* (2015) 'Research of AGV positioning based on the two-dimensional Code Recognition Method', *2015 International Conference on Logistics, Informatics and Service Science, LISS 2015*. doi: 10.1109/LISS.2015.7369812.
- Li, S., Yan, J. and Li, L. (2018) 'Automated Guided Vehicle: The Direction of Intelligent Logistics', *Proceedings of the 2018 IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI 2018*. IEEE, pp. 250–255. doi: 10.1109/SOLI.2018.8476726.
- Qi, B. Y., Zhou, Y. Y. and Yang, Q. L. (2016) 'Application of AGV in intelligent logistics system', pp. 5 .-5 . doi: 10.1049/cp.2015.1527.
- Sampath, B. G., Perera, G. V. A. G. A. and Dassanayake, W. D. I. G. (2014) 'A Fuzzy Integrated Self-tuning PID Technique for Mini Robot'.
- Takaya, K. *et al.* (2016) 'Simulation Environment for Mobile Robots Testing Using ROS and Gazebo', pp. 96–101.
- Tran, H. A. M. *et al.* (2018) 'Develop of AGV Platform to Support The Arrangement of Cargo in Storehouse', *2018 24th International Conference on Automation and Computing (ICAC)*. Chinese Automation and Computing Society in the UK - CACSUK, (September), pp. 1–5. doi: 10.23919/ICoNAC.2018.8748948.